COMVAN: A Novel Data Structure for Storing Large Database for Incremental Association Mining

Archana Gupta[1], Sanjeev Jain[2], Akhilesh Tiwari[3]
[1]Research Scholar, Madhav Institute of Technology & Science, Gwalior (M.P.), India
[2]Vice-Chancellor, Shri Mata Vaishno Devi University (SMVDU), Katra, J&K- 182320, India
[3]Associate Professor, Department of CSE & IT, Madhav Institute of Technology & Science, Gwalior (M.P.),
[1]archana100gupta@gmail.com, [2]dr_sanjeevjain@yahoo.com, [3]atiwari.mits@gmail.com

***Abstract* - This paper proposes a newand efficient data structure for storing large database to support Incremental Association Mining. Incremental algorithms are used to modify the results of earlier mining to derive the results for the incremented database. Various business databases are incremental in nature and knowledge is required from the updated database to infer some decisions and predictions. At times it is needed to ignore the previous knowledge obtained from the old database. Thus incremental algorithm should be able to manipulate the inferred rules when some part of the present database is deleted. To implement algorithms for incremental mining, first of all, an efficient data structure to store the data is needed. An easiest and most understandable way of storing the data is in the relational form. But in case of incremental implementation relational form of the data will incur high complexity. Literature reveals that number of tree based data structures are proposed by different researchers in [1,2,3]. A critical analysis on the existing data structures has been performed and observed that most of the earlier data structures were not efficient. Considering the efficiency issue in the storage and handling of large data for incremental association mining, present paper introduces a novel tree-based data structure.**

***Keywords*- Association Mining, Database, Incremental Mining, Tree Based Data Structure**

1. INTRODUCTION:

In present scenario, database mining [19,20,21,22,23]has attracted many database communities due to its wide applicability to improve business strategies. Association mining techniques are very important in data mining applications. Association rules have been applied to many areasincluding outlier detection, classification, clustering,etc[8,9,10,11,12]. ARM techniques have very wide scope of applications. For Example – given a database of sales transaction of retail store, it would be interesting to discover the association among the items such that the presence of some item in the transaction imply the presence of some another items in the same transactions. These association rules are useful in framing various business strategies so as to improve the performance. It is also useful in medical field in various ways.

Many Algorithms have already been proposed for Incremental association rule mining. In incremental approach, when dataset is updated by new information then it has to preserve old information along with new information and it has to mine the knowledge from the entire database.

In many algorithms proposed by different researchers, it is needed to rescan the old database to infer the results. This approach is not efficient as database size is very huge. So we need some implementation in which there should be no need to rescan the old database again to infer results.

There are two basic approaches of association rule mining [18,16].:Apriori based method [6] and Tree based method [7,14,13,15]

Tree based methods are more suitable for incremental mining as compared to apriori based methods. The first tree based method for association rule mining is FP Tree based method[7]. FP Tree is a data structure used to store the information of database in tree form. A database is converted in tree structure and then algorithm is applied on tree data structure to infer the association knowledge. In this method there is no need to scan the original database once it got convert in tree. That is the advantage of tree based method which need scanning of original database once or twice only.

In incremental mining since database will get update time to time so accordingly FP tree needs to be modified. Thus two types of implementations are possible :

1. Rescanning of old database is needed and then it will generate the tree for the entire database.
2. No rescanning of old database is needed and it will generate the tree by updating the old tree with the new incremented information. Second approach is more efficient as it won't need rescanning of old database whose size is considerably very large. Thus, improving the overall efficiency of the system. In real life applications, databases are continuously incremented with time. The incremental mining algorithms always need to preserve the old data along with the new data to infer new knowledge. So the main challenge is to design the algorithm for association mining without any need for rescanning the old database. Many researchers have worked on the same issue by taking into consideration the same factor. [1] proposes DB-Tree and potential frequent pattern Tree (PotFP-Tree) algorithms for mining incremental association rules in updated database. [3] proposes FELINE CATS Tree and AFPIM algorithm for incremental mining. [2] proposes CAN Tree for mining frequent itemsets in incremental database

If we are able to generate the tree for the given database then the information in the database is converted in to tree form. Then the next responsibility is to design the algorithm to retrieve the information from the tree and perform association mining to infer knowledge.

Thispaper introduces **Canonical Ordered tree with Multiple VAluesNode tree or COMVAN tree** and COMVAN tree builder algorithm. Once COMVAN tree is built then apply

FEEPAMTalgorithm to access the frequent pattern from the given transactions.

2. OUTLINE OF THE PAPER

The organization of the rest of the paper is as follows: section 3 introduces the COMVAN:A novel tree based data structure for the incremental association rule mining and the algorithm to construct it. COMVAN Tree based frequent pattern mining algorithm is introduced in section 4. Section 5 discusses some additional benefits of COMVAN Tree. Finally conclusions are given in section 6.

3. COMVAN: A NOVEL TREE BASED DATA STRUCTURE

This method is based on the lexicographic order of the items. First of all, all the attributes must be arranged in the lexicographic order. Then the algorithm will read one item at a time from the record and insert the item in the tree based on the following rules :

1. Root will point to the first item in the first record..
2. Associated with every node in the tree, a list of values is there. A value at a particular position k represents the total number of records in which that item is at kthposition.
3. Every next item in the record will be added as a child of the preceding item in the record.
4. If the item is already there in the children of the preceding node then no new node will be added, only the corresponding value in the list will get increment by 1.
5. If the first item of the record is not in the children of the root, then it will search for the first item in the tree using DFS. Once item is found in the tree, then it will follow the same procedure to enter the record in the tree. If that item is not found in the tree then it will be added as a child of the root.
6. The length of the list associated with every node will depend on the level of the tree at which that node is located. The number of values in the list will be same as the number of level. The number at particular position in the list represents the total number of records having that item at this position in reverse. If the length of the list is n, then the value at the nth position represents the number of record in which the position of item is first. Similarly the value at n-1 position represents the number of records in which the position of the item is second. The value at first position represents the number of records with its nth position. So, in general the value at kth position represents the cardinality of the records where the position of that item is (n-k+1).

```
Algorithm : COMVAN Tree Builder
Input : Set of Transactions
Output :COMVAN Tree structure
Algorithm :
While (end of the database)
{
Read   one record of n itemsat a time;
If Item [1] is available at level 1 then
{
add 1 to the value at list[1];
p=1;
}
Else
search in tree for item[1] using DFS, let it is at level p and add
1 to the value at list[p] for item[1];
For k = 2 to n
{
If (item[k] is in the children of item[k-1])  then
add 1 to the value at list[p] of item[k];
Else
add a node as child of item[k-1] with label item[k] with
list[p]=1;
}
}
```

The sample transactional database for implementing the above algorithm is given in table 1.

Table 1.The example transactional database

Tid	Items bought
101	a, b, c, d, e, g
102	a, b, c, d, h
103	a, b, c
104	a, b, c, d
105	a, b
106	b, c, d, e, g
107	b, c, d, f
108	b, c, d, f
109	b, c, g
110	b, c, d
111	b, d, e
112	b, c, d
113	b, c, d
114	d, e, h
115	d, f
116	d, e
117	c
118	d
119	d, f, h
120	a, b, c, d, f, h

Figure 1 gives the novel tree structure generated with the above algorithm.

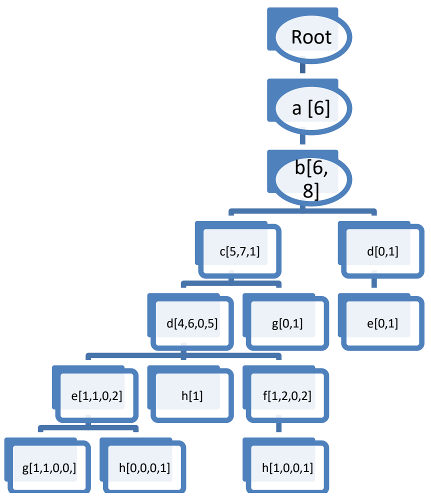

Figure 1.The Novel tree of the example database given in table 1.

4. FREQUENTPATTERNMINING WITH COMVAN TREE : (FEEPAMT)

Unlike FP-Tree, once the COMVAN tree is constructed, it can be mined repeatedly for frequent patterns with different support threshold without any need to rebuild the tree. We can perform constraint mining very easily here by using COMVAN Tree.

In COMVAN Tree, we use a list of values called List to represent the frequency of that item at different position in the set of frequent itemset. To search for the frequent itemset, traverse the tree from top to down. While traversing from top to down check for the values in the list associated with the node say m. if any of the value in the list crosses the support threshold let's say i^th^value in the list of node m i.e. list[i] for m node then add litem item m in the list of frequent itemset and go to the children of node m.

If any of the child node let's say p is having list[i]>= σ then p will be added with m in the list of frequent itemset i.e. [m,p]. repeat the process until either for any child node list[i]<σ or there is no child node.

Unlike FP Tree[7] or CATS Tree [3], we can get frequent pattern anywhere in the tree.

```
Algorithm : FEEPAMT

Input : a COMVAN Tree and required support

Output: a set of frequent pattern

Procedure : FEEPAMT (required support )
Level=0;  // Initially control will be at root
Parent = child of node at level=0;
Level++;
While ( tree is having nodes)
{
While ( no more child left for parent/ children are over)
{
I=level-1;
While (i<=level)
{
If list[i] >= σ then add the item in Frequent itemsets;
Go to the children of that node
While (no more child left)
{
if list[i]>= σ then
{
add that child in the list of frequent itemset along with parent node;
and go to the children of that child
}
else break;
}
I++;
}
Level++;
}
```

5. Discussion

In this section, we will discuss two issues: (i) the applicability of the proposed COMVAN tree in the incremental constrained mining and (ii) efficiency and memory issues regarding our COMVAN Trees.

5.1 Applicability for constrained mining

So far, we have given how to construct COMVAN tree for the given database and how efficient it is for the incremented database. However, it is important to note that COMVAN Tree has also provide some additional functionalities such as it can also be used for incremental constraint mining. Besides incremental mining, frequent pattern mining is generalized to many forms, one of which is constrained mining. The use of constraint permits user guidance that enables user exploration and and control and always lead to the effective pruning of the search space. The constrained mining provides efficient discovery of frequent patternssatisfying the user specified constraint. By applying their constraint, user can arrange the items in any order. The constraint can be any rule applied for the pruning of database for example the price of the item should be greater the 100 INR. Other constraints can be (i) furniture has high priority the kitchen items, (ii) constraint can be applied on the basis of overall sale, (iii) on the basis of stock left for a particular product, etc.

FIC Algorithm [4] and FPS Algorithm [5] have already been proposed for constraint mining but they are using FP Tree for the same. These algorithms can use COMVAN Tree for to arrange tree items in some canonical order.

5.2 Efficiency and Memory issues

By considering the above implementation of COMVAN Tree on sample database, it is observed that number of nodes required is very less in our proposed novel tree structure. In this implementation every node has a list of values associated with it. There is a special procedure to read the values. The position of the value has its own significance. We can easily construct the original database from the given tree. Since it needs lesser number of nodes , hence its memory requirement is always less comparatively. Thus it is more efficient as compared to the other methods given in [1], [2], [3].COMVAN Trees significantly reduce computation and time because they easily find mergable path.

6. Conclusion

This paper presents a novel data structure named as *COMVAN tree* and the algorithm to build it. In addition, a new mining algorithm called FEEPAMT for mining of frequent itemsets from COMVAN tree has been proposed.

It has been observed that the Apriori and FP tree based methods are not suitable for incremental mining; as they need rescanning of old data at every increment.

There are many advantages of newly proposed COMVAN tree over most of the existing tree based methods proposed in [1,2,3]. One of the main advantages is, once a COMVAN tree is constructed, frequent pattern mining with different support value can be performed without rebuilding the tree. It gives the many advantages of "Build once, Mine many". This COMVAN tree has great advantage if frequent patterns contain some common data items. Further, the proposed tree based data structure also eliminates the need to create separate tree; in case the support vary. It has also become possible to delete any transaction in easiest manner without the need to rescan the database. Second advantage is that, now it has become possible to construct the original table with the help of COMVAN tree and vice versa. Third advantage is that now it is required to scan the original database only once. Finally, the new tree based data structure allows the insertion and deletion of single transaction at a time; this makes it suitable for real time frequent mining.

In order to demonstrate the usefulness of proposed tree based data structure, implementation of COMVAN tree (to represent the transactional database in tree form) for efficiently mining the frequent itemsets using FEEPAMT algorithm has been done. During the analysis of proposed tree based data structure it has been realized that the proposed structure is not only memory efficient but also succeeded in offering high performance.

Conflict of Interest

The Author(s) declare that there is no conflict of interest regarding the publication of this manuscript.

References

[1] C. I. Ezeife and Y. Su. “Mining Incremental Association Rules withGeneralized FP-Tree”. Proceedings of the 15th Canadian Conferenceon Artificial Intelligence, May 2002.

[2] C. K. Leung, Q. I. Khan and T. Hoque.“CanTree: A Tree Structure forEfficient Incremental Mining of Frequent Patterns”, Proceedings ofthe Fifth IEEE International Conference on Data Mining (ICDM’05), 2005.

[3] W. Cheung and O.R. Zaiane , “ incremental mining of frequent patterns without candidate generation or support constraint”, Proceedings of IDEAS 2003, pp. 111-116.

[4] J. Pei, J. Han, and L.V.S. Lakshmanan. Mining frequent itemsets with convertible constraints.In Proc. ICDE 2001, pp. 433–442.

[5] C.K.-S. Leung, L.V.S. Lakshmanan, and R.T. Ng. Exploiting succinctconstraints using FP-trees. SIGKDD Explorations, 4(1), pp. 40–49, June 2002.

[6] Agrawal, R., Imielinski, T. and Swami, A. “Mining Association Rules between Sets of Items in LargeDatabase”. Proceedings of the ACM SIGMOD conference on management of data, Washington, D.C, May 26-28, 1993.

[7] Agrawal R and Srikant, R., “Fast algorithms for miningassociation rules”, Proceedings of the 1994 Int. Conf. Very LargeData Bases, pp. 487-499, Santiago, Chile, September1994.

[8] Brin, S., Motwani, R., and Silverstein, C. Beyond marketbaskets: Generalizing association rules to correlations.SIGMOD 26[2], 265-276. 1997.

[9] Antonie, M.-L. and Zaïane, O. R., Text DocumentCategorization by Term Association , IEEE ICDM'2002,pp 19-26, Maebashi City, Japan, December 9 - 12, 2002

[10] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., andHsu, M.-C. FreeSpan: Frequent pattern-projectedsequential pattern mining. ACM SIGKDD, 2000.

[11] Beil, F., Ester, M., Xu, X., Frequent Term-Based TextClustering, ACM SIGKDD, 2002

[12] Orlando, S., Palmerini, P., and Perego, R. Enhancing theApriori Algorithm for Frequent Set Counting. Proceedingsof 3rd International Conference on Data Warehousing andKnowledge Discovery. 2001.

[13] J. Pei, J. Han, and R. Mao, “Closet: An Efficient Algorithm forMining Frequent Closed Itemsets,” Proc. SIGMOD Int’l WorkshopData Mining and Knowledge Discovery, May2000.

[14] Huang, H., Wu, X., and Relue, R. Association Analysiswith One Scan of Databases. Proceedings of the 2002IEEE International Conference on Data Mining. 2002.

[15] Zaki, M. J. and Hsiao, C.-J. CHARM: An EfficientAlgorithm for Closed Itemset Mining. SIAM InternationalConference on Data Mining. 2002.

[16] A. Tiwari, R. K. Gupta and D. P. Agarwal. “A Survey on frequent Pattern Mining : Current Status and Challenges Issues”, Proceedings of the Information Technology Journal 9 (7), 2010, pp-1278-1293.

[17] Arun K. Pujari, “Data Mining Techniques”,published by UnversitiesPree (India) Private Limited, Hyderabad.

[18] Agrawal, R.,Imielinski,T.and Swami A. “Mining Association Rules between Sets of Items in LargeDatabase”. Proceedings of the ACM SIGMOD conference on management of data, Washington, D.C, May 26-28, 1993.

[19] SunithaVanamala, L. Padma Sree and S.DurgaBhavani, “Rare association rule mining for data stream”, proceedings of International IEEE Conference on CCOMVANomputer and Communications Technologies (ICCCT), pp 1-6, 2014.

[20]Sanmoy Bhattacherjee, Jayakrushna Sahooand Adrijit Goswami, “Association Rule Mining Approach in Strategy Planning for Team India in ICC World Cup 2015”, Proceedings of Second International IEEE Conference on Advances in Computing and Communication Engineering (ICACCE),Pp: 616 – 621, 2015.

[21] RajdeepKaurAulakh, “ Association Rules Mining Using Effective Algorithm: A Review”, Volume 5, Issue 3, ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering, March 2015.

[22] MeeraNarvekar and ShafaqueFatma Syed, “An optimized algorithm for association rule mining using FP tree” International Conference on Advanced Computing Technologies and Applications (ICACTA- 2015).

[23] PannawatSriratnaand PakornLeesutthipornchai, “Interesting -based association rules for highway traffic data”, Proceedings of IEEE Conference, International Computer Science and Engineering Conference (ICSEC), pp 1-6, 2015.